The quickest way I know for NFA to DFA:

1. Define $\delta(p, c) = \exists q: \text{you can get to } q \text{ by processing } c \text{ first then any } \varepsilon$-moves.

2. $\Delta(p, c) = \bigcup_{p' \in P} \delta(p', c)$

$S, F$ same as before for DFA.

Example:

$$N = \{1, 2, 3\}$$

Whenever 2, then also 3.

$$\delta(\{2, 3\}, a) = \delta(2, a) \cup \delta(3, a) = \{3\} \cup \{1\} = \{3\}$$

$$\delta(\{2, 3\}, b) = \delta(2, b) \cup \delta(3, b) = \{1\} \cup \{1, 2, 3\} = \{1, 2, 3\}$$

Now the $\varepsilon$-closure is inside.

Use Breadth First Search.

$\Delta(\{1, 3\}, b) = \delta(1, b) \cup \delta(3, b) = \{2, 3\} \cup \{2, 3\} = \{2, 3\}$

New state $M$ goes on.
Let's trace the original NFA on input $x = b b a b a a$

$S = \{12\} \cup \{2, 3\} \cup \{1, 2, 3\} \cup \{1, 2, 3\} \cup \{12\} \cup L(N)$.

$N$ and $L(N)$ have no "nirvana" condition.

Do they have a "dead" condition? No. No $\emptyset$ state in $M$.

N' =

A GNFA can't convert to DFA or even NFA directly because it can "jump" 2 or more chars in a step.

Helpful intuition: An NFA has arcs labeled $c$ when $c \in \Sigma$

or "$\emptyset$" for arcs not there.
Definition: \( \text{Regexp}(\Sigma) \) denotes the set of all legal regular expressions \( r \) over \( \Sigma \).

A GNFA is a 5-tuple \( G = (Q, \Sigma, \delta, s, F) \) when \( Q, s, F \) are as with NFAs, but
\[
\delta : (Q \times \text{Regexp}(\Sigma) \times Q) \rightarrow Q.
\]

Definition: A GNFA can "process" a string \( X \) from state \( p \) to state \( q \) if there is a sequence
\[
(q_0, W_1, q_1, W_2, q_2, \ldots, q_{m-1}, W_m, q_m)
\]
such that \( q_0 = p, q_m = q \), \( W_1 \ldots W_m = X \), and for \( 1 \leq i \leq m \), there is an instruction \( (q_{i-1}, r, q_i) \) where \( r \in \text{Regexp}(\Sigma) \) and \( W_i \in L(r) \).

There is an arc \( \gamma_{q_{i-1}} \rightarrow q_i \) such that \( r \) matches the substring of \( X \).
"Incoming + Outgoing Tracking Idea"

Eliminate a Nonaccepting State q of a GNFA.

Proof builds a bypass highway

Strategy: DON'T do text step with new S or S' yet

1) Eliminate all y \& F, q ≠ s

by bypassing all incoming edges (p, β, q) as diagrammed
then simply delete q.

2) If you have at most 3 states
Left, read off answer(s) at right:

Else, do trick of adding new F
with ε arcs to it from all accepting states.
Then eliminate all other q ≠ S. You
will get case at right with A, N = Ø.

Theorem: For every GNFA G, we can
build a regex p s.t. L(r) = L(G).